Opus Coarse Energy Predictor Notes (Round 5)

Jake Taylor (yupferris@gmail.com), 3 July 2021

I've solved it - I was simply mistaken about what the output of the filters should be. Instead of the filters outputting the *prediction*, they output the *residual* directly. Under this interpretation, everything falls into place.

According to the paper[1]/RFC[2], the 2D z-transform should be:

$$A(z_\ell, z_b) = (1 - \alpha z_\ell^{-1}) \cdot \frac{1 - z_b^{-1}}{1 - \beta z_b^{-1}}$$

I've simplified the source code (in unquant_coarse_energy in quant_bands.c in libopus 1.3.1[3]) to the following C (pseudo)code:

```
void unquant_coarse_energy(float *e, int bands) {
  float alpha = /* ... */;
  float beta = /* ... */;
  float prev = 0.0f;
  for (int b = 0; b < bands; b++) {
    float r = /* read from bitstream */;
    e[i] = alpha * e[i] + prev + r;
    prev = prev + (1 - beta) * r;
  }
}
```

The goal is simple: extract difference equations from the code, derive the corresponding z-transforms, and then solve the final z-transform which should hopefully match what the paper/RFC list.

The code is governed by the following difference equations:

$$e[\ell, b] = \alpha e[\ell - 1, b] + prev[\ell, b - 1] + r[\ell, b]$$

$$prev[\ell, b] = prev[\ell, b - 1] + (1 - \beta) r[\ell, b]$$

The corresponding z-transforms are:

$$E(z_\ell, z_b) = \alpha z_\ell^{-1} E(z_\ell, z_b) + z_b^{-1} Prev(z_\ell, z_b) + R(z_\ell, z_b)$$

$$Prev(z_\ell, z_b) = z_b^{-1} Prev(z_\ell, z_b) + (1 - \beta) R(z_\ell, z_b)$$

We expect the domain of the predictor to be e (the 2D "energy plane") and the range of the predictor to be r (the final coded residual). So, these two z-transforms should be all we need to reach the z-transform from the paper/RFC.

First, let's simplify the latter equation by isolating $Prev$:

$$Prev(z_\ell, z_b) = \frac{1 - \beta}{1 - z_b^{-1}} R(z_\ell, z_b)$$

Now, we can substitute this definition in the first equation, which leaves only E and R signals (our expected domain and range, respectively):

$$E(z_\ell, z_b) = \alpha z_\ell^{-1} E(z_\ell, z_b) + z_b^{-1} \cdot \frac{1 - \beta}{1 - z_b^{-1}} R(z_\ell, z_b) + R(z_\ell, z_b)$$

Simplifying this yields:

$$R(z_\ell, z_b) = (1 - \alpha z_\ell^{-1}) \cdot \frac{1 - z_b^{-1}}{1 - \beta z_b^{-1}} E(z_\ell, z_b)$$

Or, equivalently:

$$A(z_\ell, z_b) = (1 - \alpha z_\ell^{-1}) \cdot \frac{1 - z_b^{-1}}{1 - \beta z_b^{-1}}$$

Just as expected!

[1]https://arxiv.org/abs/1602.04845

[2]https://datatracker.ietf.org/doc/html/rfc6716#section-4.3.2

[3]https://opus-codec.org/release/stable/2019/04/12/libopus-1_3_1.html